

IN THE CLAIMS:

Please amend claims 23, 24, 31, and 32 as follows.

1. (Original) A memory management method comprising the steps of:

assigning pointers to free memory locations;

linking said pointers to one another creating a linked list of free memory locations having a beginning and an end;

assigning a free head pointer to a memory location indicating said beginning of free memory locations;

assigning a free tail pointer to a memory location indicating said end of free memory locations;

assigning an initial data pointer to said memory location assigned to said free head pointer;

assigning an end of data pointer to a last data memory location;

assigning said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein said next memory location indicates said beginning of free memory locations.

2. (Original) The method as recited in claim 1 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer is assigned to said memory location assigned to said initial data pointer.

3. (Original) The method as recited in claim 1 wherein when data to be saved is larger than one memory location the method comprises the steps of:

assigning a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

4. (Original) The method as recited in claim 1 further comprising the steps of:

linking said memory location assigned to said free tail pointer to said memory location assigned to said initial data pointer when said data is to be indicated as free memory; and

assigning said free tail pointer to said last data memory location assigned to said end of data pointer.

5. (Original) A memory management method comprising the steps of:

assigning pointers to free memory locations;

linking said pointers to one another creating a linked list of free memory locations having a beginning and an end;

assigning a free head pointer to a memory location indicating said beginning of free memory locations;

assigning a free tail pointer to a memory location indicating said end of free memory locations;

linking said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory; and

assigning said free tail pointer to said last data memory location assigned to said end of data pointer.

6. (Original) The method as recited in claim 5 further comprising the steps of:

assigning an initial data pointer to said memory location assigned to said free head pointer;

assigning an end of data pointer to a last data memory location;

assigning said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein said next memory location indicates said beginning of free memory locations.

7. (Original) The method as recited in claim 6 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer is assigned to said memory location assigned to said initial data pointer.

8. (Original) The method as recited in claim 6 wherein when data to be saved is larger than one memory location the method comprises the steps of:

assigning a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

9. (Original) A memory management system comprising:

a pointer assignor that assigns pointers to free memory locations;

a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end;

a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations;

a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations;

an initial data pointer assignor that assigns an initial data pointer to said memory location assigned to said free head pointer; and

an end of data pointer assignor that assigns an end of data pointer to a last data memory location;

wherein said free head pointer assignor assigns said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein said next memory location indicates said beginning of free memory locations.

10. (Original) The system as recited in claim 9 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer assignor assigns said end of data pointer to said memory location assigned to said initial data pointer.

11. (Original) The system as recited in claim 9 wherein when data to be saved is larger than one memory location the system further comprises:

a data assignor that assigns a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

12. (Original) The system as recited in claim 9 wherein said linker links said memory location assigned to said free tail pointer to said memory location assigned to said initial data pointer when said data is to be indicated as free memory; and

said assignor assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

13. (Original) A memory management system comprising:

a pointer assignor that assigns pointers to free memory locations;

a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end;

a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations; and

a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations;

wherein said linker links said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory; and

said free tail pointer assignor assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

14. (Original) The system as recited in claim 13 further comprising:

an initial data pointer assignor that assigns an initial data pointer to said memory location assigned to said free head pointer; and

an end of data pointer assignor that assigns an end of data pointer to a last data memory location;

wherein said free head pointer assignor assigns said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein said next memory location indicates said beginning of free memory locations.

15. (Original) The system as recited in claim 14 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer assignor assigns said end of data pointer to said memory location assigned to said initial data pointer.

16. (Original) The system as recited in claim 14 wherein when data to be saved is larger than one memory location, a data assignor assigns a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

17. (Original) A memory management system comprising:

- a pointer assignor means for assigning pointers to free memory locations;
- a linker means for linking said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end;
- a free head pointer assignor means for assigning a free head pointer to a memory location indicating said beginning of said linked list of free memory locations;
- a free tail pointer assignor means for assigning a free tail pointer to a memory location indicating said end of said linked list of free memory locations;
- an initial data pointer assignor means for assigning an initial data pointer to said memory location assigned to said free head pointer; and
- an end of data pointer assignor means for assigning an end of data pointer to a last data memory location;

wherein said free head pointer assignor means assigns said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein said next memory location indicates said beginning of free memory locations.

18. (Original) The system as recited in claim 17 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer assignor means assigns said end of data pointer to said memory location assigned to said initial data pointer.

19. (Original) The system as recited in claim 17 wherein when data to be saved is larger than one memory location the system further comprises:

a data assignor means for assigning a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

20. (Original) The system as recited in claim 17 wherein said linker means links said memory location assigned to said free tail pointer to said memory location assigned to said initial data pointer when said data is to be indicated as free memory; and

said assignor means assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

21. (Original) A memory management system comprising:

a pointer assignor means for assigning pointers to free memory locations;

a linker means for linking said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end;

a free head pointer assignor means for assigning a free head pointer to a memory location indicating said beginning of said linked list of free memory locations; and

a free tail pointer assignor means for assigning a free tail pointer to a memory location indicating said end of said linked list of free memory locations;

wherein said linker means links said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory; and

said free tail pointer assignor means assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

22. (Original) The system as recited in claim 21 further comprising:

an initial data pointer assignor means for assigning an initial data pointer to said memory location assigned to said free head pointer; and

an end of data pointer assignor means for assigning an end of data pointer to a last data memory location;

wherein said free head pointer assignor means assigns said free head pointer to a next memory location linked to said last data memory location assigned to said end of

data pointer, wherein said next memory location indicates said beginning of free memory locations.

23. (Currently Amended) The system as recited in claim ~~21~~ 22 wherein when data to be saved is only large enough to occupy one memory location said end of data pointer assignor means assigns said end of data pointer to said memory location assigned to said initial data pointer.

24. (Currently Amended) The system as recited in claim ~~21~~ 22 wherein when data to be saved is larger than one memory location, a data assignor means assigns a sufficient number of memory locations linked to said memory location assigned to said initial data pointer.

25. (Original) A memory management device comprising:
a pointer assignor that assigns pointers to free memory locations of a memory, said pointer assignor in communication with said memory;
a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, said linker in communication with said memory;

a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, said free head pointer assignor in communication with said memory;

a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations, said free tail pointer assignor in communication with said memory;

an initial data packet pointer assignor that assigns an initial data packet pointer to said memory location assigned to said free head pointer, said initial data packet pointer assignor in communication with said memory; and

an end of data packet pointer assignor that assigns an end of data packet pointer to a last data memory location in said memory, said end of data packet pointer assignor in communication with said memory;

wherein said free head pointer assignor assigns said free head pointer to a next memory location linked to said last data packet memory location assigned to said end of data packet pointer, wherein said next memory location indicates said beginning of free memory locations.

26. (Original) The device as recited in claim 25 wherein when a single data packet is to be saved said end of data packet pointer assignor assigns said end of data packet pointer to said memory location assigned to said initial data packet pointer.

27. (Original) The device as recited in claim 25 wherein the device further comprises:

a data packet assignor that assigns a sufficient number of memory locations linked to said memory location assigned to said initial data packet pointer, said data packet assignor in communication with said memory.

28. (Original) The device as recited in claim 25 wherein said linker links said memory location assigned to said free tail pointer to said memory location assigned to said initial data packet pointer when said data packet is to be indicated as free memory; and

said assignor assigns said free tail pointer to said last data packet memory location assigned to said end of data packet pointer.

29. (Original) A memory management device comprising:

a pointer assignor that assigns pointers to free memory locations of a memory, said pointer assignor in communication with said memory;

a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, said linker in communication with said memory;

a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, said free head pointer assignor in communication with said memory; and

a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations, said free tail pointer assignor in communication with said memory;

wherein said linker links said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory; and

said free tail pointer assignor assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

30. (Original) The device as recited in claim 29 further comprising:

an initial data packet pointer assignor that assigns an initial data packet pointer to said memory location assigned to said free head pointer, said initial data packet pointer assignor in communication with said memory; and

an end of data packet pointer assignor that assigns an end of data packet pointer to a last data packet memory location, said end of data packet pointer assignor in communication with said memory;

wherein said free head pointer assignor assigns said free head pointer to a next memory location linked to said last data packet memory location assigned to said end of

data packet pointer, wherein said next memory location indicates said beginning of free memory locations.

31. (Currently Amended) The device as recited in claim ~~29~~ 30 wherein when a single data packet is to be saved said end of data packet pointer assignor assigns said end of data packet pointer to said memory location assigned to said initial data packet pointer.

32. (Currently Amended) The device as recited in claim ~~29~~ 30 wherein when more than one data packet is to be saved, a data packet assignor assigns a sufficient number of memory locations linked to said memory location assigned to said initial data packet pointer, said data packet assignor in communication with said memory.